Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals of DAAAM International

SIMPLE AND CHEAP MICROCONTROLLER KIT FOR STUDENTS

DOLINAY, J[an]; DOSTALEK, P[etr] & VASEK, V[ladimir]

Abstract: *This paper presents simple evaluation kit for students who learn microcontroller programming. This is second generation of such a kit which was modified based on experience with the previous version. It is very easy to make and cheap which allows students to build it at home and experiment with it. It is based on Atmel AVR Mega microcontroller and is connected to PC via serial port.*
Key words: *microcontroller, AVR, learning kit, ATMega*

1. INTRODUCTION

Microcontrollers are nowadays present in almost every device around us and their usage increases. With this increase of usage increases also the need for qualified experts able to program them. It is the task of universities and colleges to prepare such experts, software engineers, able to program microcontrollers efficiently and correctly.

At our faculty we developed a simple and cheap microcontroller tool which can be used in lessons of microcontroller programming (Dolinay at al., 2007). Our aim was to make it possible for a student to work with microcontroller (MCU) not only at the lessons but also at home, to obtain some real experience with the methods used for programming and so on.

Based on practical experience with this tool we designed a new version with improved features. This second generation of the simple microcontroller kit is described in this article. The improvements are in the new type of MCU used, which is better suited for this task and also in the way of connecting the kit to the computer, which is also simplified. Despite the new features the kit still remains simple and affordable.

As for the question why we strive for cheap solution, which may arise, the motivation for cheap solution is not driven by the inability to obtain more sophisticated equipment for our lessons but by two other factors: First, it is the desire of some students to further investigate the topic. For such a student, buying the same equipment that is used in lessons may be too expensive and borrowing it from school is not always possible. The second reason is simplicity. The devices used in lessons are professional products which actually hide the basic principle in their sophisticated design. In a device with many parts it may be hard to even find the microcontroller itself. It is desirable, at least at the beginning, to allow the students to see how things are made. That is why simple board containing just the MCU and few other parts is helpful. Better yet, if the student can make the board himself.

2. DESIGN CONSIDERATIONS

As already mentioned, the aim of our design is to provide students with easy-to-build platform for developing microcontroller applications. This tool should be used mainly at home for the student's own projects but it should also refer to the lessons at school. In the lessons students first learn programming in assembly and then in C. We use evaluation kit M68EVB908GB60 with Freescale HCS08 microcontroller, which is 8 bit MCU with FLASH and RAM memory (***, 2009). This kit is equipped with LCD display; several push switches, LED diodes and even a buzzer. We have also developed several extension modules which can be attached to this kit, for example, 7-segment, 4 digit multiplexed display and keyboard, DC motor drive and simple heating plant (Dolinay et al., 2007). The development kit is well suited for the purpose of learning MCU programming, but the advantage of many interesting peripherals brings also the disadvantage of complexity. This makes it hard to see the hardware principles and the cost of such a kit is beyond student's budget. Moreover, the student may soon want to create his own microcontroller application and for such a home project the surface mounted MCU with over 60 pins is not the best choice. The kit itself is too complicated to provide much inspiration without good experience in electronics. For student with little experience interested in simple way to use an MCU in his/her project the kit described here has been designed.

2.1 Choice of the microcontroller

For the first version of the kit we limited our choice to three brands of microcontrollers that we have personal experience with and which belong among the most used: Freescale, Atmel and Microchip. Luckily, the principles are very much the same for all microcontrollers so the choice is not critical. The preference of particular brand is affected by factors such as personal experience, price, features and development tools available.

For microcontroller manufacturers the hobby-users are probably not the most important customers, but for our aim of providing simple and affordable kit for students which they could make at home, the popularity in hobby market segment is an important clue.

Freescale microcontrollers, which we use at lessons, are widely used in professional applications - in automotive industry, etc. For hobby use they are not so popular, perhaps because it is not easy to obtain them (situation in the Czech Republic). Atmel microcontrollers (***, 2010) are very popular among the hobby users because of low price, good features and performance and availability of free software including full featured C compiler (AVR studio, WinAVR). PIC microcontrollers are very popular in hobby area also, but it seems they are loosing the battle to Atmel at present due to lack of good free development tools.

For the first generation of the kit we used ATTiny 2313 MCU; for the second generation we chose ATMega8. This microcontroller has bigger memory, (8 KB of flash and 1 KB of RAM compared to 2 KB and 128 bytes in Tiny2313) for very similar price. Also, there are versions with even more memory available with the same pin configuration (ATMega168 with 16 KB and ATMega328 with even 32 KB of flash).

2.2 Connection to PC

Another important decision for the kit design was the means of connecting the kit with the PC on which the programs are developed. This choice is a compromise between the comfort of usage and the price of such a tool. The best option nowadays is obviously connection via USB, but for our purpose

of easy-to-make and affordable solution this has disadvantage of requiring more parts which make the kit more expensive. Also the kit could become too complex to understand for beginners and the home building of this kind of device could have doubtful effectiveness. It could be easier (and in the end cheaper) to buy a professional programmer or development kit. For the above reasons we chose programming via In-System-Programming (ISP) interface for the first kit. This kind of programming interface has the disadvantage of requiring special cable and computer with parallel or serial port. This is a problem especially with laptops, which now often have only USB ports. The simple solution of using USB to serial converter did not work for the ISP programmer communication on neither of several converters we used, probably due to the fact that the converter does not emulate all the functionality of a hardware serial port. Because of this, the first generation kit could only be used on computers with serial or parallel port, which proved to be a major limitation to students. To remove this limitation we chose a different approach for the second generation of the kit. It is now programmed via bootloader in the MCU which communicates with the PC via serial line. Unlike the ISP programming this communication uses the serial port in a standard way and there is no problem with USB to serial converters. So the new kit can be connected to any laptop using any cheap USB to serial converter. The disadvantage of this design is the fact, that the MCU has to have the bootloader programmed before it can be used in the kit. The bootloader can be loaded into the MCU either at school or by some friend who has a programmer for AVR MCUs. In total this limitation seems easier to overcome that the requirement for a hardware serial port.

3. KIT DESCRIPTION

For our kit we use Atmel AVR microcontroller Mega8, but it can be simply replaced by pin-compatible Mega168 or Mega328 which have more memory. The MCU is clocked by internal oscillator, eliminating the need for a crystal.

The schematic of the kit is depicted in figure 1. Besides the socket for MCU there is integrated voltage regulator 7805 to obtain 5V for powering the kit, level converter MAX232 for connecting the TTL logic levels of the MCU serial interface to the serial port on PC; then 2 LEDs and 2 push switches to allow some inputs/outputs for the MCU. A trim is used to generate variable voltage input for experiments with A/D converter. It should be noted that the peripherals are not connected directly to the MCU, but to a DIP8 socket. The MCU pins are made available using sockets also. This way any peripheral can be connected to any pin of the MCU using a piece of wire.

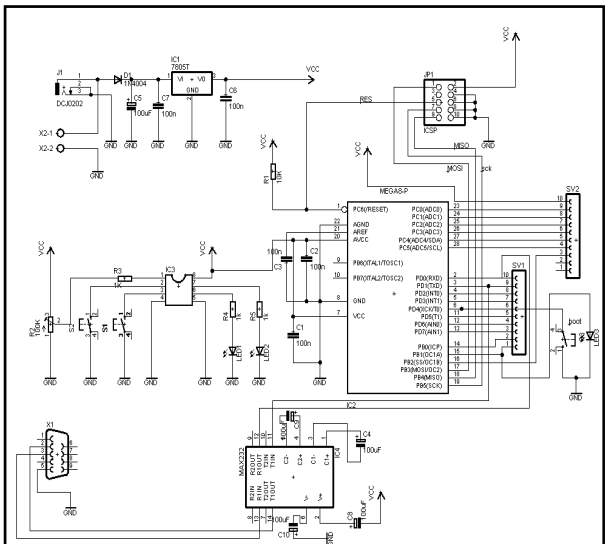

Fig. 1. Schematics of the kit

For loading program into the MCU a bootloader is used. It is custom version of bootloader published in (Isa, 2010). To put the MCU into programming mode, the pin PD4 must be tied to ground when the MCU is powered up. The bootloader is then ready to receive new program. This is achieved by holding down a push switch and signalled by LED. The bootloader occupies only 512 words of memory, so there is still plenty of space for user programs. The program is sent into the MCU using avrdude program, which is part of the installation of AVR Studio. The program can be started from command line as follows:

avrdude -p m8 -c avr109 -b 4800 -P com1 -U program.hex

The m8 is the type of the MCU, in our case Mega8. For Mega168 it is m168 etc.

The –b option specifies the communication speed, here 4800 bd and the com1 is the serial port used for the communication. The program.hex is the filename of the hex file which contains the program to be programmed into the MCU memory. It is possible to create a simple .bat file containing this command and execute it from the command line to flash the program into MCU after building the project in AVR Studio.

To program the bootloader into a new MCU, some programmer is required, for example the ISP programmer for parallel port as used in the first version of the kit. The fuses can be then set to that the bootloader is protected from overwriting so the student may experiment with the kit without worries.

4. CONCLUSION

This paper presents very simple learning kit for students of microcontroller programming courses. It is a second generation of the kit, improved over the first one by new MCU and different way of connection with PC which allows using it on computers without serial or parallel port. The purpose of this kit is to allow students experimenting with their own MCU applications. It is not intended to be the main development platform for learning MCU programming - for this the school is equipped with professional learning kits. This development kit may be considered tool for home projects. It provides starting point which is guaranteed to work and opens the way for student's own designs. The purpose of this project is to show the way of applying microcontroller in a real application and allow easy way of programming it. In future tutorials and sample applications for the kit should be created to provide better starting conditions for student's projects.

5. ACKNOWLEDGEMENTS

This work was supported by research project MSM 7088352102. This support is very gratefully acknowledged.

6. REFERENCES

Dolinay, J.; Dostalek, P. & Vasek, V. (2007). Educational models for lessons of microcontroller programming, *Proceedings of 11th International research/expert conference TMT 2007*, pp. 1447-1450, ISBN 978-9958-617-34-8, Tunisia, September 2007, Hammamet.

Isa Jiri (2010). Bootloader for ATmega8-board, *Available from:* http://artax.karlin.mff.cuni.cz/~isa_j1am/projects /bootloader-for-atmega8-board, *Accessed*: 2010-06-22

*** (2009) http://www.freescale.com – Freescale 8-bit Microcontrollers, *Accessed on*: 2009-12-12

*** (2010) http://www.atmel.com, Atmel 8 and 32-bit MCUs, *Accessed on:* 2010-05-12

Dolinay, J.; Dostalek, P. & Vasek, V. (2008). Simple learning kit for microcontroller programming lessons, *Proceedings of the 19th International DAAAM Symposium*, pp. 417-418, ISBN 1726-9679, Slovakia, October 2008, DAAAM International, Trnava